Theoretische Grundlagen der Informatik

Vorlesung am 10.01.2023

Torsten Ueckerdt | 10. Januar 2023

Letzte Vorlesung: Approximationsalgorithmen

Optimierungsproblem Π (bzw. Optimalwertproblem)

Instanz I $\rightsquigarrow$ optimale Lösungen haben Wert $\mathrm{OPT}(I)$

Algorithmus $\mathcal{A}$ $\rightsquigarrow$ liefert Lösung mit Wert $\mathcal{A}(I)$

Absolute Approximation		**Relative Approximation**
"additiver Fehler"	**vs.**	"multiplikativer Fehler"
$\mathcal{A}(I) \leq \mathrm{OPT}(I) + K$		$\mathcal{A}(I) \leq \mathrm{OPT}(I) \cdot K$

(hier für Minimierungsproblem Π)

Approximation mit relativer Gütegarantie

Definition.

Sei Π ein Optimierungsproblem. Ein polynomialer Algorithmus $\mathcal{A}$, der für jedes $I \in D_\Pi$ einen Wert $\mathcal{A}(I)$ liefert mit $R_\mathcal{A}(I) \leq K$, wobei $K \geq 1$ eine Konstante, und

$$\mathcal{R}_\mathcal{A}(I) := \begin{cases} \frac{\mathcal{A}(I)}{\mathrm{OPT}(I)} & \text{falls } \Pi \text{ Minimierungsproblem} \\[2ex] \frac{\mathrm{OPT}(I)}{\mathcal{A}(I)} & \text{falls } \Pi \text{ Maximierungsproblem} \end{cases}$$

heißt Approximationsalgorithmus mit relativer Gütegarantie oder relativer Approximationsalgorithmus.

- Einige, aber nicht alle $\mathcal{NP}$-schweren Optimierungsprobleme erlauben einen relativen Approximationsalgorithmus.

Relative Approximation: Genereller Ansatz

Bei Minimierungsproblemen

- Wir wollen $\mathcal{R}_{\mathcal{A}}(I) = \frac{\mathcal{A}(I)}{\text{OPT}(I)} \leq K$, also $\mathcal{A}(I) \leq K \cdot \text{OPT}(I)$.
- Wir brauchen:
  - eine **obere** Schranke für $\mathcal{A}(I)$ — "$\mathcal{A}$ ist gut"
  - eine **untere** Schranke für $\text{OPT}(I)$ — "viel besser geht es nicht"

$$\mathcal{A}(I) \leq X \text{ und } \text{OPT}(I) \geq Y \quad \Longrightarrow \quad \mathcal{A}(I) \leq X = \frac{X \cdot Y}{Y} \leq \frac{X}{Y} \cdot \text{OPT}(I)$$

Relative Approximation: Genereller Ansatz

Bei Minimierungsproblemen

- Wir wollen $\mathcal{R}_{\mathcal{A}}(I) = \frac{\mathcal{A}(I)}{\mathrm{OPT}(I)} \leq K$, also $\mathcal{A}(I) \leq K \cdot \mathrm{OPT}(I)$.
- Wir brauchen:
  - eine **obere** Schranke für $\mathcal{A}(I)$ "$\mathcal{A}$ ist gut"
  - eine **untere** Schranke für $\mathrm{OPT}(I)$ "viel besser geht es nicht"

$$\mathcal{A}(I) \leq X \text{ und } \mathrm{OPT}(I) \geq Y \quad \Longrightarrow \quad \mathcal{A}(I) \leq X = \frac{X \cdot Y}{Y} \leq \frac{X}{Y} \cdot \mathrm{OPT}(I)$$

Bei Maximierungsproblemen

- Wir wollen $\mathcal{R}_{\mathcal{A}}(I) = \frac{\mathrm{OPT}(I)}{\mathcal{A}(I)} \leq K$, also $\mathcal{A}(I) \geq \frac{1}{K} \cdot \mathrm{OPT}(I)$.
- Wir brauchen:
  - eine **untere** Schranke für $\mathcal{A}(I)$ "$\mathcal{A}$ ist gut"
  - eine **obere** Schranke für $\mathrm{OPT}(I)$ "viel besser geht es nicht"

$$\mathcal{A}(I) \geq X \text{ und } \mathrm{OPT}(I) \leq Y \quad \Longrightarrow \quad \mathcal{A}(I) \geq X = \frac{X \cdot Y}{Y} \geq \frac{X}{Y} \cdot \mathrm{OPT}(I)$$

Beispiel: Greedy-Algorithmus für KNAPSACK

Idee:

Bevorzuge Elemente mit günstigem Kosten-pro-Gewicht Verhältnis, also hoher Kostendichte.

$\rightsquigarrow$ Es werden der Reihe nach so viele Elemente wie möglich mit absteigender Kostendichte in die Lösung aufgenommen.

- Berechne die Kostendichten $p_i := \frac{c_i}{w_i}$ für $i = 1, \ldots, n$
- Sortiere nach Kostendichten und indiziere: $p_1 \geq p_2 \geq \cdots \geq p_n$
- Dies kann in Zeit $\mathcal{O}(n \log n)$ geschehen.
- Für $i = 1$ bis n setze $x_i := \left\lfloor \frac{W}{w_i} \right\rfloor$ und $W := W - \left\lfloor \frac{W}{w_i} \right\rfloor \cdot w_i$.

Die Laufzeit dieses Algorithmus ist in $\mathcal{O}(n \log n)$.

KNAPSACK-Instanz

$M = \{1, \ldots, n\}$,
Kosten $c_1, \ldots, c_n$,
Gewichte $w_1, \ldots, w_n$,
Gesamtgewicht W.

Beispiel: Greedy-Algorithmus für KNAPSACK

- Berechne die Kostendichten $p_i := \frac{c_i}{w_i}$ für $i = 1, \ldots, n$
- Sortiere nach Kostendichten und indiziere: $p_1 \geq p_2 \geq \cdots \geq p_n$
- Dies kann in Zeit $\mathcal{O}(n \log n)$ geschehen.
- Für $i = 1$ bis n setze $x_i := \left\lfloor \frac{W}{w_i} \right\rfloor$ und $W := W - \left\lfloor \frac{W}{w_i} \right\rfloor \cdot w_i$.

KNAPSACK-Instanz

$M = \{1, \ldots, n\}$,
Kosten $c_1, \ldots, c_n$,
Gewichte $w_1, \ldots, w_n$,
Gesamtgewicht W.

Satz.

Der Greedy-Algorithmus $\mathcal{A}$ erfüllt $\mathcal{R}_{\mathcal{A}}(I) \leq 2$ für alle Instanzen I.

Bei Maximierungsproblemen

- Wir wollen $\mathcal{R}_{\mathcal{A}}(I) = \frac{\mathrm{OPT}(I)}{\mathcal{A}(I)} \leq K$, also $\mathcal{A}(I) \geq \frac{1}{K} \cdot \mathrm{OPT}(I)$.
- Wir brauchen:
  - eine **untere** Schranke für $\mathcal{A}(I)$ — “$\mathcal{A}$ ist gut”
  - eine **obere** Schranke für $\mathrm{OPT}(I)$ — “viel besser geht es nicht”

Beispiel: Greedy-Algorithmus für KNAPSACK

- Berechne die Kostendichten $p_i := \frac{c_i}{w_i}$ für $i = 1, \dots, n$
- Sortiere nach Kostendichten und indiziere: $p_1 \geq p_2 \geq \cdots \geq p_n$
- Dies kann in Zeit $\mathcal{O}(n \log n)$ geschehen.
- Für $i = 1$ bis n setze $x_i := \left\lfloor \frac{W}{w_i} \right\rfloor$ und $W := W - \left\lfloor \frac{W}{w_i} \right\rfloor \cdot w_i$.

KNAPSACK-Instanz

$M = \{1, \dots, n\}$,
Kosten $c_1, \dots, c_n$,
Gewichte $w_1, \dots, w_n$,
Gesamtgewicht W.

Beweis: (O.B.d.A. sei $w_1 \leq W$.)

Eine untere Schranke für $\mathcal{A}(I)$: $\mathcal{A}(I) \geq c_1 \cdot x_1 = c_1 \cdot \left\lfloor \frac{W}{w_1} \right\rfloor$

Eine obere Schranke für $\mathrm{OPT}(I)$: $\mathrm{OPT}(I) \leq p_1 \cdot W = \frac{c_1}{w_1} \cdot W$

Beispiel: Greedy-Algorithmus für KNAPSACK

- Berechne die Kostendichten $p_i := \frac{c_i}{w_i}$ für $i = 1, \ldots, n$
- Sortiere nach Kostendichten und indiziere: $p_1 \geq p_2 \geq \cdots \geq p_n$
- Dies kann in Zeit $\mathcal{O}(n \log n)$ geschehen.
- Für $i = 1$ bis n setze $x_i := \left\lfloor \frac{W}{w_i} \right\rfloor$ und $W := W - \left\lfloor \frac{W}{w_i} \right\rfloor \cdot w_i$.

KNAPSACK-Instanz

$M = \{1, \ldots, n\}$,
Kosten $c_1, \ldots, c_n$,
Gewichte $w_1, \ldots, w_n$,
Gesamtgewicht W.

Beweis: (O.B.d.A. sei $w_1 \leq W$.)

Eine untere Schranke für $\mathcal{A}(I)$: $\mathcal{A}(I) \geq c_1 \cdot x_1 = c_1 \cdot \left\lfloor \frac{W}{w_1} \right\rfloor$

Eine obere Schranke für $\mathrm{OPT}(I)$: $\mathrm{OPT}(I) \leq p_1 \cdot W = \frac{c_1}{w_1} \cdot W$

Dann gilt

$$\mathrm{OPT}(I) \leq c_1 \cdot \frac{W}{w_1} \leq c_1 \cdot \left(\left\lfloor \frac{W}{w_1} \right\rfloor + 1 \right) \leq 2 \cdot c_1 \cdot \left\lfloor \frac{W}{w_1} \right\rfloor \leq 2 \cdot \mathcal{A}(I) .$$

Also $\mathcal{R}_{\mathcal{A}}(I) \leq 2$.

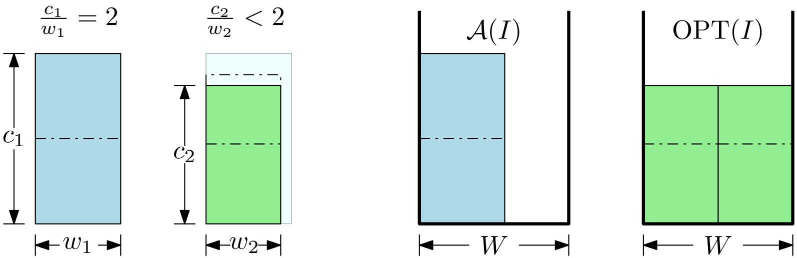
Grenzen für den Greedy-Algorithmus

Karlsruher Institut für Technologie

Bemerkung: Die Schranke $\mathcal{R}_{\mathcal{A}}(I) \leq 2$ ist in gewissem Sinne scharf.

- Sei $n = 2,\ w_2 = w_1 - 1,\ c_1 = 2 \cdot w_1,\ c_2 = 2 \cdot w_2 - 1,\ W = 2 \cdot w_2$.
- Dann ist $\frac{c_1}{w_1} = 2$ aber $\frac{c_2}{w_2} < 2$.
- Es gilt $\mathcal{A}(I) = c_1$ und $\mathrm{OPT}(I) = 2c_2$, also

$$\frac{\mathrm{OPT}(I)}{\mathcal{A}(I)} = \frac{2c_2}{c_1} = \frac{4w_1 - 6}{2w_1} \longrightarrow 2 \qquad \text{für } w_1 \to \infty$$



Mit $\mathcal{NP}$-schweren Problemen umgehen: Übersicht

Karlsruher Institut für Technologie

Pseudopolynomiale Algorithmen

- Laufzeit: $\text{poly}(|I|, \max(I))$ — polynomial in $|I|$ und $\max(I)$
- $\mathcal{A}(I) = \text{OPT}(I)$ — optimal, kein Fehler

Absolute Approximationsalgorithmen

- Laufzeit: $\text{poly}(|I|)$ — polynomial
- $|\mathcal{A}(I) - \text{OPT}(I)| \leq K$ — absoluter Fehler
  - $\mathcal{A}(I) \leq \text{OPT}(I) + K$ bei Minimierungsproblem
  - $\mathcal{A}(I) \geq \text{OPT}(I) - K$ bei Maximierungsproblem

Relative Approximationsalgorithmen

- Laufzeit: $\text{poly}(|I|)$ — polynomial
- $\mathcal{A}(I) \leq K \cdot \text{OPT}(I)$ bei Minimierungsproblem — relativer Fehler
- $\mathcal{A}(I) \geq \frac{1}{K} \cdot \text{OPT}(I)$ bei Maximierungsproblem

Bemerkung:
Es gibt noch weitere Ansätze.

Bestmögliche Approximationsgüte

Definition.

Zu einem polynomialen Approximationsalgorithmus $\mathcal{A}$ sei

$$\mathcal{R}_{\mathcal{A}}^{\infty} := \inf \left\{ r \geq 1 : \begin{array}{c} \text{es gibt ein } N_0 > 0 \text{, so dass } \mathcal{R}_{\mathcal{A}}(I) \leq r \\ \text{für alle } I \text{ mit } \mathrm{OPT}(I) \geq N_0 \end{array} \right\}$$

Beispiel:

- Angenommen $\mathcal{A}(I) \leq K \cdot \mathrm{OPT}(I) + 3$ für alle I.
- Dann ist $\mathcal{R}_{\mathcal{A}}(I) = K + 1$ für $\mathrm{OPT}(I) = 3$
  und $\mathcal{R}_{\mathcal{A}}(I) = K + \frac{1}{2}$ für $\mathrm{OPT}(I) = 6$, usw.
- Wir haben aber $\mathcal{R}_{\mathcal{A}}^{\infty} = K$.

Erinnerung:

$\mathcal{R}_{\mathcal{A}}(I) = \frac{\mathcal{A}(I)}{\mathrm{OPT}(I)}$
für Minimierungsproblem

$\mathcal{R}_{\mathcal{A}}(I) = \frac{\mathrm{OPT}(I)}{\mathcal{A}(I)}$
für Maximierungsproblem

Bestmögliche Approximationsgüte

Definition.

Zu einem polynomialen Approximationsalgorithmus $\mathcal{A}$ sei

$$\mathcal{R}_{\mathcal{A}}^{\infty} := \inf \left\{ r \geq 1 : \begin{array}{c} \text{es gibt ein } N_0 > 0 \text{, so dass } \mathcal{R}_{\mathcal{A}}(I) \leq r \\ \text{für alle } I \text{ mit } \mathrm{OPT}(I) \geq N_0 \end{array} \right\}$$

Erinnerung:

$\mathcal{R}_{\mathcal{A}}(I) = \frac{\mathcal{A}(I)}{\mathrm{OPT}(I)}$
für Minimierungsproblem

$\mathcal{R}_{\mathcal{A}}(I) = \frac{\mathrm{OPT}(I)}{\mathcal{A}(I)}$
für Maximierungsproblem

Beispiel:

- Angenommen $\mathcal{A}(I) \leq K \cdot \mathrm{OPT}(I) + 3$ für alle I.
- Dann ist $\mathcal{R}_{\mathcal{A}}(I) = K + 1$ für $\mathrm{OPT}(I) = 3$
  und $\mathcal{R}_{\mathcal{A}}(I) = K + \frac{1}{2}$ für $\mathrm{OPT}(I) = 6$, usw.
- Wir haben aber $\mathcal{R}_{\mathcal{A}}^{\infty} = K$.

$$\mathcal{A}(I) \leq \mathrm{OPT}(I) + K \text{ für alle } I \quad \Longrightarrow \quad \mathcal{R}_{\mathcal{A}}^{\infty} = 1$$

Übersicht

relative Approximation

$\mathcal{A}(I) \leq K \cdot \mathrm{OPT}(I)$ bzw. $\mathcal{R}_{\mathcal{A}}^{\infty} \leq K$

poly$(|I|)$

absolute Approx.

$|\mathcal{A}(I) - \mathrm{OPT}(I)| \leq K$

poly$(|I|)$

Übersicht

relative Approximation

$\mathcal{A}(I) \leq K \cdot \mathrm{OPT}(I)$ bzw. $\mathcal{R}_{\mathcal{A}}^{\infty} \leq K$

poly($|I|$)

absolute Approx.

$|\mathcal{A}(I) - \mathrm{OPT}(I)| \leq K$

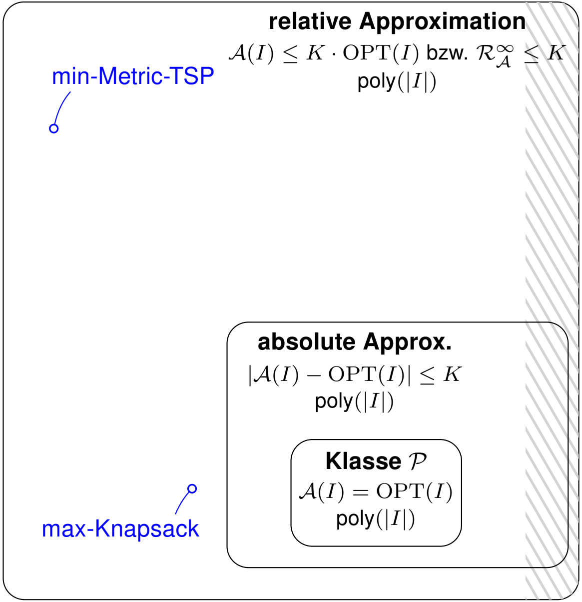
poly($|I|$)

Klasse $\mathcal{P}$

$\mathcal{A}(I) = \mathrm{OPT}(I)$

poly($|I|$)

Übersicht


relative Approximation
$\mathcal{A}(I) \leq K \cdot \mathrm{OPT}(I)$ bzw. $\mathcal{R}_{\mathcal{A}}^{\infty} \leq K$
poly($|I|$)
min-Metric-TSP
absolute Approx.
$|\mathcal{A}(I) - \mathrm{OPT}(I)| \leq K$
poly($|I|$)
Klasse $\mathcal{P}$
$\mathcal{A}(I) = \mathrm{OPT}(I)$
poly($|I|$)
max-Knapsack

Metrisches TSP

Karlsruher Institut für Technologie

Optimalwertproblem min-METRIC-TSP

Gegeben: vollständiger Graph $G = (V, E)$,
Gewichtsfunktion $c\colon E \longrightarrow \mathbb{Q}$
mit $c(u, w) \leq c(u, v) + c(v, w)$ für alle $u, v, w \in V$

Aufgabe: Minimiere die Länge bezüglich c von einer Tour zu G.

Satz.

Für das Optimalwertproblem min-METRIC-TSP existiert ein relativer Approximationsalgorithmus $\mathcal{A}$ mit $\mathcal{R}^{\infty}_{\mathcal{A}} \leq 2$.

Bemerkung:

- Es gilt sogar $\mathcal{R}_{\mathcal{A}}(I) \leq 2$ für alle Instanzen I.

2-Approximation von min-METRIC-TSP

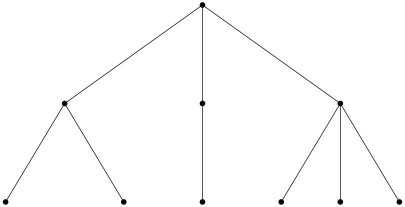
Beweis.

- Sei $I = (G = (V, E), c)$ eine Instanz von min-METRIC-TSP.

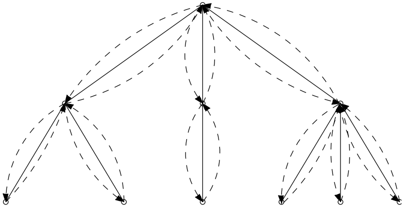
Betrachte folgenden Algorithmus $\mathcal{A}$:

1. Berechne einen *MST* (Minimum Spanning Tree) von G.
2. Wähle einen beliebigen Knoten w als Wurzel.
3. Durchlaufe den *MST* in einer Tiefensuche mit Startpunkt w
4. **Dies liefert:** Tour T mit Start- und Endpunkt w, die jede Kante genau zweimal durchläuft.
5. Konstruiere entlang T eine abgekürzte Tour T', indem bereits besuchte Knoten übersprungen werden und die Tour T' beim nächsten unbesuchten Knoten fortgesetzt wird.
6. **Ergebnis:** $\mathcal{A}(I) = c(T') = \sum_{e \in T'} c(e)$

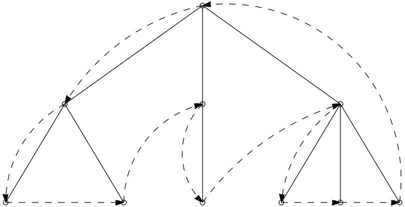
2-Approximation von min-METRIC-TSP



(a) MST eines Graphen



(b) Tiefensuch-Tour durch den MST



(c) TSP-Tour als abgekürzte Tiefensuch-Tour

2-Approximation von min-METRIC-TSP

Approximationsgüte:

Laufzeit:

- $\mathcal{O}(n^2)$ für $n = |V|$.
- Das ist poly($|I|$).

2-Approximation von min-METRIC-TSP

Approximationsgüte:

Bei Minimierungsproblemen

- Wir wollen $\mathcal{R}_{\mathcal{A}}(I) = \frac{\mathcal{A}(I)}{\mathrm{OPT}(I)} \leq K$, also $\mathcal{A}(I) \leq K \cdot \mathrm{OPT}(I)$.
- Wir brauchen:
  - eine **obere** Schranke für $\mathcal{A}(I)$ "$\mathcal{A}$ ist gut"
  - eine **untere** Schranke für $\mathrm{OPT}(I)$ "viel besser geht es nicht"

Laufzeit:

- $\mathcal{O}(n^2)$ für $n = |V|$.
- Das ist poly$(|I|)$.

2-Approximation von min-METRIC-TSP

Approximationsgüte:

Bei Minimierungsproblemen

- Wir wollen $\mathcal{R}_{\mathcal{A}}(I) = \frac{\mathcal{A}(I)}{\mathrm{OPT}(I)} \leq K$, also $\mathcal{A}(I) \leq K \cdot \mathrm{OPT}(I)$.
- Wir brauchen:
  - eine **obere** Schranke für $\mathcal{A}(I)$ "$\mathcal{A}$ ist gut"
  - eine **untere** Schranke für $\mathrm{OPT}(I)$ "viel besser geht es nicht"

- Obere Schranke für $\mathcal{A}(I)$: $\mathcal{A}(I) = c(T') \leq c(T) = 2 \cdot c(MST)$
- Untere Schranke für $\mathrm{OPT}(I)$: $\mathrm{OPT}(I) \geq c(MST)$.

  Denn: Eine TSP-Tour kann als ein aufspannender Baum plus eine zusätzliche Kante betrachtet werden. Und *MST* ist ein kürzester aufspannender Baum.

Laufzeit:

- $\mathcal{O}(n^2)$ für $n = |V|$.
- Das ist poly($|I|$).

2-Approximation von min-METRIC-TSP

Approximationsgüte:

- Obere Schranke für $\mathcal{A}(I)$: $\mathcal{A}(I) = c(T') \leq c(T) = 2 \cdot c(MST)$
- Untere Schranke für $\mathrm{OPT}(I)$: $\mathrm{OPT}(I) \geq c(MST)$.

  Denn: Eine TSP-Tour kann als ein aufspannender Baum plus eine zusätzliche Kante betrachtet werden. Und *MST* ist ein kürzester aufspannender Baum.

- Insgesamt erhält man

$$\mathcal{R}_{\mathcal{A}}(I) \leq \frac{\text{obere Schranke}}{\text{untere Schranke}} = \frac{2 \cdot c(MST)}{c(MST)} = 2.$$

  Das heißt $\mathcal{A}(I) \leq 2 \cdot c(MST) \leq 2 \cdot \mathrm{OPT}(I)$.

- $\rightsquigarrow \mathcal{R}_{\mathcal{A}}^{\infty} \leq 2$.

Laufzeit:

- $\mathcal{O}(n^2)$ für $n = |V|$.
- Das ist $\mathrm{poly}(|I|)$.

Bemerkungen zur Approximierbarkeit

min-METRIC-TSP

- Wir haben eine 2-Approximation mit $\mathcal{O}(n^2)$ Laufzeit gesehen.
- **Christofides 1976:** Es gibt eine 1.5-Approximation mit $\mathcal{O}(n^3)$ Laufzeit.
- **Karpinski et al. 2015:** Es gibt keine $\frac{123}{122}$-Approximation mit polynomialer Laufzeit.
- **Karlin et al. 2021:** Es gibt eine α-Approximation mit $\alpha < 1.5$ mit polynomialer Laufzeit.

Bemerkungen zur Approximierbarkeit

min-METRIC-TSP

- Wir haben eine 2-Approximation mit $\mathcal{O}(n^2)$ Laufzeit gesehen.
- **Christofides 1976:** Es gibt eine 1.5-Approximation mit $\mathcal{O}(n^3)$ Laufzeit.
- **Karpinski et al. 2015:** Es gibt keine $\frac{123}{122}$-Approximation mit polynomialer Laufzeit.
- **Karlin et al. 2021:** Es gibt eine α-Approximation mit $\alpha < 1.5$ mit polynomialer Laufzeit.

max-KNAPSACK

- Wir haben eine 2-Approximation mit $\mathcal{O}(n \log n)$ Laufzeit gesehen.

  $\rightsquigarrow$ Greedy Algorithmus
- Es gibt eine 1.5-Approximation mit $\mathcal{O}(n^3)$ Laufzeit.

Bemerkungen zur Approximierbarkeit

min-METRIC-TSP

- Wir haben eine 2-Approximation mit $\mathcal{O}(n^2)$ Laufzeit gesehen.
- **Christofides 1976:** Es gibt eine 1.5-Approximation mit $\mathcal{O}(n^3)$ Laufzeit.
- **Karpinski et al. 2015:** Es gibt keine $\frac{123}{122}$-Approximation mit polynomialer Laufzeit.
- **Karlin et al. 2021:** Es gibt eine α-Approximation mit $\alpha < 1.5$ mit polynomialer Laufzeit.

max-KNAPSACK

- Wir haben eine 2-Approximation mit $\mathcal{O}(n \log n)$ Laufzeit gesehen.

  $\rightsquigarrow$ Greedy Algorithmus
- Es gibt eine 1.5-Approximation mit $\mathcal{O}(n^3)$ Laufzeit.
- Es gibt eine 1.25-Approximation mit $\mathcal{O}(n^3)$ Laufzeit.

Bemerkungen zur Approximierbarkeit

min-METRIC-TSP

- Wir haben eine 2-Approximation mit $\mathcal{O}(n^2)$ Laufzeit gesehen.
- **Christofides 1976:** Es gibt eine 1.5-Approximation mit $\mathcal{O}(n^3)$ Laufzeit.
- **Karpinski et al. 2015:** Es gibt keine $\frac{123}{122}$-Approximation mit polynomialer Laufzeit.
- **Karlin et al. 2021:** Es gibt eine α-Approximation mit $\alpha < 1.5$ mit polynomialer Laufzeit.

max-KNAPSACK

- Wir haben eine 2-Approximation mit $\mathcal{O}(n \log n)$ Laufzeit gesehen.

  $\rightsquigarrow$ Greedy Algorithmus
- Es gibt eine 1.5-Approximation mit $\mathcal{O}(n^3)$ Laufzeit.
- Es gibt eine 1.25-Approximation mit $\mathcal{O}(n^3)$ Laufzeit.
- Es gibt eine 1.0001-Approximation mit $\mathcal{O}(n^3)$ Laufzeit.

Bemerkungen zur Approximierbarkeit

min-METRIC-TSP

- Wir haben eine 2-Approximation mit $\mathcal{O}(n^2)$ Laufzeit gesehen.
- **Christofides 1976:** Es gibt eine 1.5-Approximation mit $\mathcal{O}(n^3)$ Laufzeit.
- **Karpinski et al. 2015:** Es gibt keine $\frac{123}{122}$-Approximation mit polynomialer Laufzeit.
- **Karlin et al. 2021:** Es gibt eine α-Approximation mit $\alpha < 1.5$ mit polynomialer Laufzeit.

max-KNAPSACK

- Wir haben eine 2-Approximation mit $\mathcal{O}(n \log n)$ Laufzeit gesehen.

  $\rightsquigarrow$ Greedy Algorithmus
- Es gibt eine 1.5-Approximation mit $\mathcal{O}(n^3)$ Laufzeit.
- Es gibt eine 1.25-Approximation mit $\mathcal{O}(n^3)$ Laufzeit.
- Es gibt eine 1.0001-Approximation mit $\mathcal{O}(n^3)$ Laufzeit.

$\boxed{\rightsquigarrow \textbf{FPTAS}}$

Approximationsschemata

Karlsruher Institut für Technologie

Definition.

Ein Approximationsschema für ein Optimierungsproblem Π ist eine Familie von Algorithmen $\{\mathcal{A}_\varepsilon \mid \varepsilon > 0\}$, so dass für alle $\varepsilon > 0$:

- $\mathcal{R}_{\mathcal{A}_\varepsilon} \leq 1 + \varepsilon$

Ein PTAS ist ein Approximationsschema bei dem

- die Laufzeit von $\mathcal{A}_\varepsilon$ polynomial in $|I|$ ist.

Ein FPTAS ist ein Approximationsschema bei dem

- die Laufzeit von $\mathcal{A}_\varepsilon$ polynomial in $|I|$ und $\frac{1}{\varepsilon}$ ist.

(**F**)**PTAS** steht für

(**F**ully)
Polynomial
Time
Approximation
Scheme

Approximationsschemata

Karlsruher Institut für Technologie

Definition.

Ein Approximationsschema für ein Optimierungsproblem Π ist eine Familie von Algorithmen $\{\mathcal{A}_\varepsilon \mid \varepsilon > 0\}$, so dass für alle $\varepsilon > 0$:

- $\mathcal{R}_{\mathcal{A}_\varepsilon} \leq 1 + \varepsilon$ $\rightsquigarrow$ beliebig gute Approximation

Ein PTAS ist ein Approximationsschema bei dem

- die Laufzeit von $\mathcal{A}_\varepsilon$ polynomial in $|I|$ ist. $\rightsquigarrow$ poly($|I|$)

Ein FPTAS ist ein Approximationsschema bei dem

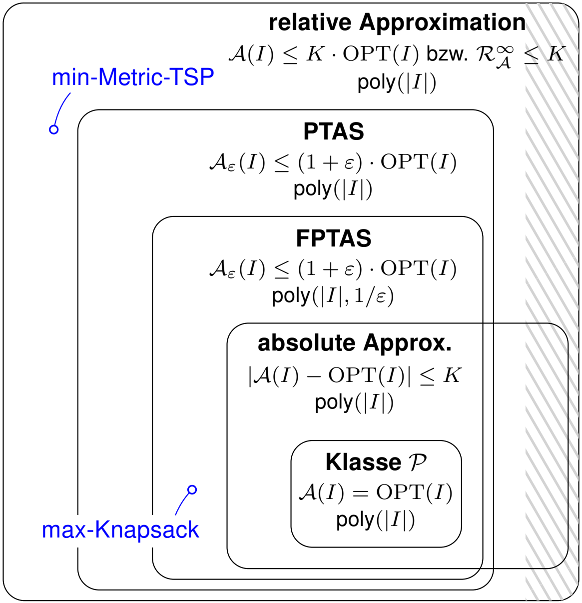
- die Laufzeit von $\mathcal{A}_\varepsilon$ polynomial in $|I|$ und $\frac{1}{\varepsilon}$ ist. $\rightsquigarrow$ poly($|I|, 1/\varepsilon$)

(**F**)**PTAS** steht für

(**F**ully)
Polynomial
Time
Approximation
Scheme

- Ein **PTAS** erlaubt Laufzeiten von $\mathcal{O}(n^{1/\varepsilon})$. $n = |I|$
  z.B. $\mathcal{O}(n)$ für 2-Approx., $\mathcal{O}(n^2)$ für 1.5-Approx., $\mathcal{O}(n^4)$ für 1.25-Approx.
- Ein **FPTAS** erlaubt Laufzeiten von $\mathcal{O}(\frac{1}{\varepsilon} \cdot n)$.
  z.B. $\mathcal{O}(n)$ für 2-Approx., $\mathcal{O}(n)$ für 1.5-Approx., $\mathcal{O}(n)$ für 1.25-Approx.

Übersicht


relative Approximation
$\mathcal{A}(I) \leq K \cdot \mathrm{OPT}(I)$ bzw. $\mathcal{R}_{\mathcal{A}}^{\infty} \leq K$
poly$(|I|)$
min-Metric-TSP
PTAS
$\mathcal{A}_\varepsilon(I) \leq (1+\varepsilon) \cdot \mathrm{OPT}(I)$
poly$(|I|)$
FPTAS
$\mathcal{A}_\varepsilon(I) \leq (1+\varepsilon) \cdot \mathrm{OPT}(I)$
poly$(|I|, 1/\varepsilon)$
absolute Approx.
$|\mathcal{A}(I) - \mathrm{OPT}(I)| \leq K$
poly$(|I|)$
Klasse $\mathcal{P}$
$\mathcal{A}(I) = \mathrm{OPT}(I)$
poly$(|I|)$
max-Knapsack

Ein FPTAS für max-KNAPSACK

Optimierungsproblem max-KNAPSACK

Gegeben: Eine endliche Menge M,
eine Gewichtsfunktion $w\colon M \longrightarrow \mathbb{N}_0$,
eine Kostenfunktion $c\colon M \longrightarrow \mathbb{N}_0$, $W \in \mathbb{N}$.

Aufgabe: Maximiere $c(M')$ für eine Teilmenge M' von M mit $w(M') \leq W$.

Ein FPTAS für max-KNAPSACK

Optimierungsproblem max-KNAPSACK

Gegeben: Eine endliche Menge M,
eine Gewichtsfunktion $w\colon M \longrightarrow \mathbb{N}_0$,
eine Kostenfunktion $c\colon M \longrightarrow \mathbb{N}_0$, $W \in \mathbb{N}$.

Aufgabe: Maximiere $c(M')$ für eine Teilmenge M' von M mit $w(M') \leq W$.

Unser Vorgehen: Variiere den pseudopolynomialen Algorithmus $\mathcal{A}$ aus letzter Vorlesung.

$\rightsquigarrow$ Laufzeit: $\mathcal{O}(|M| \cdot c(M))$

- Für $\varepsilon > 0$ entwerfe $(1 + \varepsilon)$-Approximation $\mathcal{A}_\varepsilon$ wie folgt:
  1. Bei Eingabe $I = (M, w, c, W)$, berechne ein k aus $|I|$, $\max(I)$ und ε.
  2. Skaliere Kostenfunktion $c'(i) = \lfloor c(i)/k \rfloor$.
  3. Berechne $\mathcal{A}$ auf Eingabe (M, w, c', W).
- Beweise: Laufzeit von $\mathcal{A}_\varepsilon = \text{poly}(|I|, \frac{1}{\varepsilon})$ und $\mathcal{R}_{\mathcal{A}_\varepsilon} \leq 1 + \varepsilon$.

Ein pseudopolynomialer, optimaler Algorithmus für max-KNAPSACK

Für $i \in M$, $r \leq c(M)$ berechne

$$w_r^i := \min\{w(M') \mid M' \subseteq \{1, \ldots, i\}, c(M') = r\}.$$

- **Initialisierung**
  Für $i = 1, \ldots, |M|$ setze $w_0^i := 0$
- **Berechung**
  Für $r = 1, \ldots, c(M)$ und $i = 1, \ldots, |M|$ setze
  $$w_r^i := \min\left\{w_{r-c(i)}^{i-1} + w(i), w_r^{i-1}\right\}$$
- **Ausgabe** $\mathcal{A}(I) := \max\{r \mid w_r^{|M|} \leq W\} = \mathrm{OPT}(I)$

Ein pseudopolynomialer, optimaler Algorithmus für max-Knapsack

Für $i \in M$, $r \leq c(M)$ berechne

$$w_r^i := \min\{w(M') \mid M' \subseteq \{1, \ldots, i\}, c(M') = r\}.$$

- **Initialisierung**
  Für $i = 1, \ldots, |M|$ setze $w_0^i := 0$
- **Berechung**
  Für $r = 1, \ldots, c(M)$ und $i = 1, \ldots, |M|$ setze
  $$w_r^i := \min\left\{w_{r-c(i)}^{i-1} + w(i), w_r^{i-1}\right\}$$
- **Ausgabe** $\mathcal{A}(I) := \max\{r \mid w_r^{|M|} \leq W\} = \mathrm{OPT}(I)$

$\rightsquigarrow$ **Laufzeit:** in $\mathcal{O}(|M| \cdot c(M))$.
$\rightsquigarrow$ **Lösung:** optimal, d.h. $\mathcal{A}(I) = \mathrm{OPT}(I)$.
$\rightsquigarrow$ Optimaler pseudopolynomialer Algorithmus $\mathcal{A}$.

Ein FPTAS für max-KNAPSACK

- Bezeichne $\mathcal{A}$ den vorigen pseudopolynomialen Algorithmus für KNAPSACK mit Laufzeit $\mathcal{O}(|M| \cdot c(M))$.

Definiere Algorithmus $\mathcal{A}_\varepsilon$ für $\varepsilon > 0$:

1. Bei Eingabe $I = (M, w, c, W)$, berechne

$$c_{\max} := \max\{c(i) \mid i \in M\} \quad \text{und} \quad k := \frac{c_{\max}}{\left(\frac{1}{\varepsilon} + 1\right) \cdot |M|}$$

2. Betrachte die skalierte Instanz I_k mit $c'(i) := \left\lfloor \frac{c(i)}{k} \right\rfloor$ für alle $i \in M$.
3. Berechne $\mathcal{A}$ mit Eingabe $I_k = (M, w, c', W)$.

Ein FPTAS für max-KNAPSACK

- Bezeichne $\mathcal{A}$ den vorigen pseudopolynomialen Algorithmus für KNAPSACK mit Laufzeit $\mathcal{O}(|M| \cdot c(M))$.

Definiere Algorithmus $\mathcal{A}_\varepsilon$ für $\varepsilon > 0$:

1. Bei Eingabe $I = (M, w, c, W)$, berechne
$$c_{\max} := \max\{c(i) \mid i \in M\} \quad \text{und} \quad k := \frac{c_{\max}}{\left(\frac{1}{\varepsilon} + 1\right) \cdot |M|}$$
2. Betrachte die skalierte Instanz I_k mit $c'(i) := \left\lfloor \frac{c(i)}{k} \right\rfloor$ für alle $i \in M$.
3. Berechne $\mathcal{A}$ mit Eingabe $I_k = (M, w, c', W)$.

Satz.

$\mathcal{R}_{\mathcal{A}_\varepsilon}(I) \leq 1 + \varepsilon$ für alle $I \in D_\Pi$ und die Laufzeit von $\mathcal{A}_\varepsilon$ ist in $\mathcal{O}(|I|^3 \cdot \frac{1}{\varepsilon})$ für alle $\varepsilon > 0$, d.h. $\{\mathcal{A}_\varepsilon \mid \varepsilon > 0\}$ ist ein FPTAS für max-KNAPSACK.

Ein FPTAS für max-KNAPSACK

Satz.

$\mathcal{R}_{\mathcal{A}_\varepsilon}(I) \leq 1 + \varepsilon$ für alle $I \in D_\Pi$ und die Laufzeit von $\mathcal{A}_\varepsilon$ ist in $\mathcal{O}(|I|^3 \cdot \frac{1}{\varepsilon})$ für alle $\varepsilon > 0$,
d.h. $\{\mathcal{A}_\varepsilon \mid \varepsilon > 0\}$ ist ein FPTAS für max-KNAPSACK.

Beweis:

Die Laufzeit von $\mathcal{A}_\varepsilon$ ist $\mathcal{O}(|M| \cdot c'(M))$ wobei

$$c'(M) = \sum_{i=1}^{|M|} \left\lfloor \frac{c(i)}{k} \right\rfloor \leq \sum_{i=1}^{|M|} \frac{c_i}{k} \leq |M| \cdot \frac{c_{\max}}{k} = \left(\frac{1}{\varepsilon} + 1\right) |M|^2.$$

Also ist die Laufzeit von $\mathcal{A}_\varepsilon$ in $\mathcal{O}(|M|^3 \cdot \frac{1}{\varepsilon})$ für alle $\varepsilon > 0$.

Ein FPTAS für max-KNAPSACK

Satz.

$\mathcal{R}_{\mathcal{A}_\varepsilon}(I) \leq 1 + \varepsilon$ für alle $I \in D_\Pi$ und die Laufzeit von $\mathcal{A}_\varepsilon$ ist in $\mathcal{O}(|I|^3 \cdot \frac{1}{\varepsilon})$ für alle $\varepsilon > 0$,
d.h. $\{\mathcal{A}_\varepsilon \mid \varepsilon > 0\}$ ist ein FPTAS für max-KNAPSACK.

▹ Für die Abschätzung $\mathrm{OPT}(I) \leq (1+\varepsilon) \cdot \mathcal{A}_\varepsilon(I)$.

Bei Maximierungsproblemen

- Wir wollen $\mathcal{R}_\mathcal{A}(I) = \frac{\mathrm{OPT}(I)}{\mathcal{A}(I)} \leq K$, also $\mathcal{A}(I) \geq \frac{1}{K} \cdot \mathrm{OPT}(I)$.
- Wir brauchen:
  - eine **untere** Schranke für $\mathcal{A}(I)$ — "$\mathcal{A}$ ist gut"
  - eine **obere** Schranke für $\mathrm{OPT}(I)$ — "viel besser geht es nicht"

Ein FPTAS für max-KNAPSACK

Karlsruher Institut für Technologie

Satz.

$\mathcal{R}_{\mathcal{A}_\varepsilon}(I) \leq 1 + \varepsilon$ für alle $I \in D_\Pi$ und die Laufzeit von $\mathcal{A}_\varepsilon$ ist in $\mathcal{O}(|I|^3 \cdot \frac{1}{\varepsilon})$ für alle $\varepsilon > 0$,
d.h. $\{\mathcal{A}_\varepsilon \mid \varepsilon > 0\}$ ist ein FPTAS für max-KNAPSACK.

▹ Für die Abschätzung $\mathrm{OPT}(I) \leq (1 + \varepsilon) \cdot \mathcal{A}_\varepsilon(I)$.

Bei Maximierungsproblemen

- Wir wollen $\mathcal{R}_{\mathcal{A}}(I) = \frac{\mathrm{OPT}(I)}{\mathcal{A}(I)} \leq K$, also $\mathcal{A}(I) \geq \frac{1}{K} \cdot \mathrm{OPT}(I)$.
- Wir brauchen:
  - eine **untere** Schranke für $\mathcal{A}(I)$ — "$\mathcal{A}$ ist gut"
  - eine **obere** Schranke für $\mathrm{OPT}(I)$ — "viel besser geht es nicht"

Wenn M^* optimal für I, also $\mathrm{OPT}(I) = c(M^*)$, dann

$$\mathrm{OPT}(I_k) \geq c'(M^*) = \sum_{i \in M^*} \left\lfloor \frac{c(i)}{k} \right\rfloor \geq \sum_{i \in M^*} \left(\frac{c(i)}{k} - 1 \right) \geq \frac{c(M^*)}{k} - |M|$$

Ein FPTAS für max-KNAPSACK

Karlsruher Institut für Technologie

Satz.

$\mathcal{R}_{\mathcal{A}_\varepsilon}(I) \leq 1 + \varepsilon$ für alle $I \in D_\Pi$ und die Laufzeit von $\mathcal{A}_\varepsilon$ ist in $\mathcal{O}(|I|^3 \cdot \frac{1}{\varepsilon})$ für alle $\varepsilon > 0$, d.h. $\{\mathcal{A}_\varepsilon \mid \varepsilon > 0\}$ ist ein FPTAS für max-KNAPSACK.

▹ Für die Abschätzung $\mathrm{OPT}(I) \leq (1+\varepsilon) \cdot \mathcal{A}_\varepsilon(I)$.

Wenn M^* optimal für I, also $\mathrm{OPT}(I) = c(M^*)$, dann

$$\mathrm{OPT}(I_k) \geq c'(M^*) = \sum_{i \in M^*} \left\lfloor \frac{c(i)}{k} \right\rfloor \geq \sum_{i \in M^*} \left(\frac{c(i)}{k} - 1 \right) \geq \frac{c(M^*)}{k} - |M|$$

- Eine obere Schranke für $\mathrm{OPT}(I)$:

$$\mathcal{A}_\varepsilon(I) \geq k \cdot \mathcal{A}(I_k) = k \cdot \mathrm{OPT}(I_k) \overset{!}{\geq} c(M^*) - k \cdot |M| = \mathrm{OPT}(I) - k \cdot |M|$$

Also $\boxed{\mathrm{OPT}(I) \leq \mathcal{A}_\varepsilon(I) + k \cdot |M|}$

Ein FPTAS für max-KNAPSACK

Satz.

$\mathcal{R}_{\mathcal{A}_\varepsilon}(I) \leq 1 + \varepsilon$ für alle $I \in D_\Pi$ und die Laufzeit von $\mathcal{A}_\varepsilon$ ist in $\mathcal{O}(|I|^3 \cdot \frac{1}{\varepsilon})$ für alle $\varepsilon > 0$,
d.h. $\{\mathcal{A}_\varepsilon \mid \varepsilon > 0\}$ ist ein FPTAS für max-KNAPSACK.

- Für die Abschätzung $\mathrm{OPT}(I) \leq (1+\varepsilon) \cdot \mathcal{A}_\varepsilon(I)$.
  - Eine obere Schranke für $\mathrm{OPT}(I)$:
    $$\mathcal{A}_\varepsilon(I) \geq k \cdot \mathcal{A}(I_k) = k \cdot \mathrm{OPT}(I_k) \overset{!}{\geq} c(M^*) - k \cdot |M| = \mathrm{OPT}(I) - k \cdot |M|$$
    Also $\boxed{\mathrm{OPT}(I) \leq \mathcal{A}_\varepsilon(I) + k \cdot |M|}$
  - Eine untere Schranke für $\mathcal{A}_\varepsilon(I)$:
    $$\mathcal{A}_\varepsilon(I) \geq \mathrm{OPT}(I) - k \cdot |M| \geq c_{\max} - k \cdot |M|$$
    Mit der Definition von k also $\boxed{\mathcal{A}_\varepsilon(I) \geq k \cdot |M| \cdot (1/\varepsilon)}$

Ein FPTAS für max-KNAPSACK

Satz.

$\mathcal{R}_{\mathcal{A}_\varepsilon}(I) \leq 1 + \varepsilon$ für alle $I \in D_\Pi$ und die Laufzeit von $\mathcal{A}_\varepsilon$ ist in $\mathcal{O}(|I|^3 \cdot \frac{1}{\varepsilon})$ für alle $\varepsilon > 0$, d.h. $\{\mathcal{A}_\varepsilon \mid \varepsilon > 0\}$ ist ein FPTAS für max-KNAPSACK.

▹ Für die Abschätzung $\text{OPT}(I) \leq (1+\varepsilon) \cdot \mathcal{A}_\varepsilon(I)$.

- Eine obere Schranke für $\text{OPT}(I)$:

  $$\mathcal{A}_\varepsilon(I) \geq k \cdot \mathcal{A}(I_k) = k \cdot \text{OPT}(I_k) \overset{!}{\geq} c(M^*) - k \cdot |M| = \text{OPT}(I) - k \cdot |M|$$

  Also $\boxed{\text{OPT}(I) \leq \mathcal{A}_\varepsilon(I) + k \cdot |M|}$

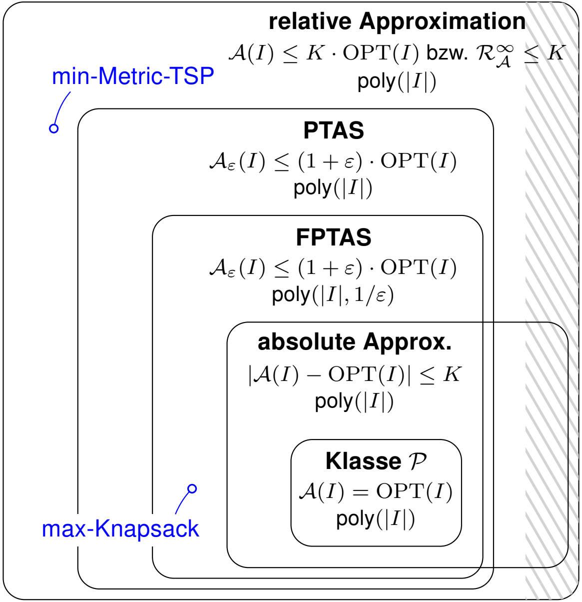
- Eine untere Schranke für $\mathcal{A}_\varepsilon(I)$:

  $$\mathcal{A}_\varepsilon(I) \geq \text{OPT}(I) - k \cdot |M| \geq c_{\max} - k \cdot |M|$$

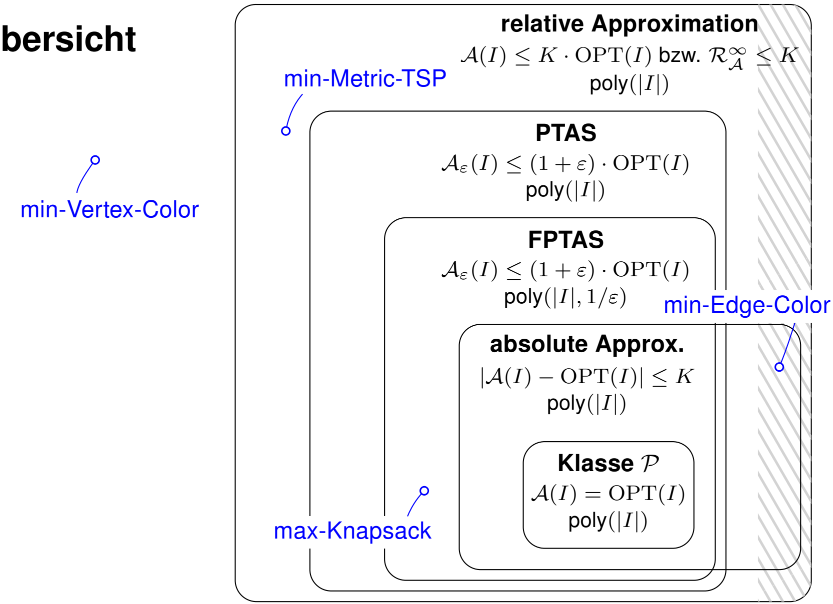
  Mit der Definition von k also $\boxed{\mathcal{A}_\varepsilon(I) \geq k \cdot |M| \cdot (1/\varepsilon)}$

- Zusammen: $\text{OPT}(I) \leq \mathcal{A}_\varepsilon(I) + k \cdot |M| \leq \mathcal{A}_\varepsilon(I) + \varepsilon \cdot \mathcal{A}_\varepsilon(I) = (1+\varepsilon) \cdot \mathcal{A}_\varepsilon(I)$

Übersicht


relative Approximation
$\mathcal{A}(I) \leq K \cdot \mathrm{OPT}(I)$ bzw. $\mathcal{R}_{\mathcal{A}}^{\infty} \leq K$
poly($|I|$)
min-Metric-TSP
PTAS
$\mathcal{A}_\varepsilon(I) \leq (1+\varepsilon) \cdot \mathrm{OPT}(I)$
poly($|I|$)
FPTAS
$\mathcal{A}_\varepsilon(I) \leq (1+\varepsilon) \cdot \mathrm{OPT}(I)$
poly($|I|, 1/\varepsilon$)
absolute Approx.
$|\mathcal{A}(I) - \mathrm{OPT}(I)| \leq K$
poly($|I|$)
Klasse $\mathcal{P}$
$\mathcal{A}(I) = \mathrm{OPT}(I)$
poly($|I|$)
max-Knapsack

Übersicht



relative Approximation
$\mathcal{A}(I) \leq K \cdot \mathrm{OPT}(I)$ bzw. $\mathcal{R}_{\mathcal{A}}^{\infty} \leq K$
poly$(|I|)$
PTAS
$\mathcal{A}_{\varepsilon}(I) \leq (1+\varepsilon) \cdot \mathrm{OPT}(I)$
poly$(|I|)$
FPTAS
$\mathcal{A}_{\varepsilon}(I) \leq (1+\varepsilon) \cdot \mathrm{OPT}(I)$
poly$(|I|, 1/\varepsilon)$
absolute Approx.
$|\mathcal{A}(I) - \mathrm{OPT}(I)| \leq K$
poly$(|I|)$
Klasse $\mathcal{P}$
$\mathcal{A}(I) = \mathrm{OPT}(I)$
poly$(|I|)$
min-Metric-TSP
min-Vertex-Color
min-Edge-Color
max-Knapsack

min-VERTEX-COLOR und min-EDGE-COLOR

Karlsruher Institut für Technologie

Optimierungsproblem min-VERTEX-COLOR

Gegeben: Graph $G = (V, E)$

Aufgabe: Färbe die Knoten in V mit möglichst wenig Farben, so dass je zwei adjazente Knoten verschiedene Farben besitzen.

Optimierungsproblem min-EDGE-COLOR

Gegeben: Graph $G = (V, E)$

Aufgabe: Färbe die Kanten in E mit möglichst wenig Farben, so dass je zwei adjazente Kanten verschiedene Farben besitzen.

Beide Entscheidungsprobleme “höchstens drei Farben” sind $\mathcal{NP}$-schwer.

Nicht-Existenz eines FPTAS

Optimierungsproblem min-Vertex-Color

Gegeben: Graph $G = (V, E)$

Aufgabe: Färbe die Knoten in V mit möglichst wenig Farben, so dass je zwei adjazente Knoten verschiedene Farben besitzen.

Satz.

Sei Π ein $\mathcal{NP}$-schweres Optimierungsproblem mit

- $\mathrm{OPT}(I) \in \mathbb{N}$ für alle $I \in D_\Pi$, und
- es existiert ein Polynom q mit $\mathrm{OPT}(I) < q(|I|)$ für alle $I \in D_\Pi$.

Falls $\mathcal{P} \neq \mathcal{NP}$, so gibt es kein FPTAS $\{\mathcal{A}_\varepsilon \mid \varepsilon > 0\}$ für Π.

Nicht-Existenz eines FPTAS

Satz.

Sei Π ein $\mathcal{NP}$-schweres Optimierungsproblem mit

- $\mathrm{OPT}(I) \in \mathbb{N}$ für alle $I \in D_\Pi$, und
- es existiert ein Polynom q mit $\mathrm{OPT}(I) < q(|I|)$ für alle $I \in D_\Pi$.

Falls $\mathcal{P} \neq \mathcal{NP}$, so gibt es kein FPTAS $\{\mathcal{A}_\varepsilon \mid \varepsilon > 0\}$ für Π.

Beweis: (O.B.d.A. für ein Maximierungsproblem Π)

- Angenommen $\{\mathcal{A}_\varepsilon \mid \varepsilon > 0\}$ sei ein FPTAS für Π.
- Wir konstruieren optimalen, polynomialen Algorithmus $\mathcal{A}$ für Π:
  1. Bei Eingabe $I \in D_\Pi$, berechne ein $\varepsilon_0 \leq \frac{1}{q(|I|)}$.
  2. Gebe $\mathcal{A}_{\varepsilon_0}(I)$ zurück. (Berechne Algorithmus $\mathcal{A}_{\varepsilon_0}$ auf Eingabe I.)

Laufzeit von $\mathcal{A}_{\varepsilon_0}$ ist $\mathrm{poly}(|I|, \frac{1}{\varepsilon_0}) = \mathrm{poly}(|I|)$, da $\frac{1}{\varepsilon_0} = q(|I|) = \mathrm{poly}(|I|)$.

Nicht-Existenz eines FPTAS

Satz.

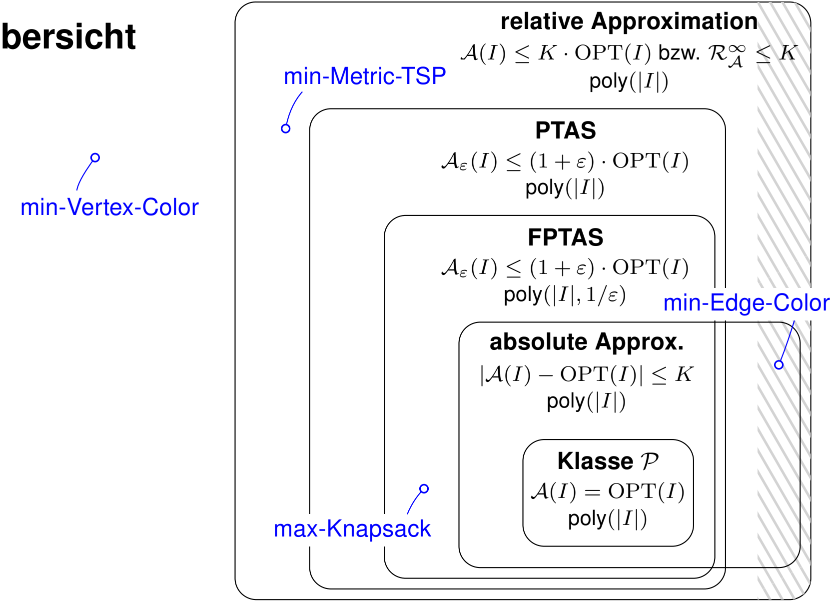
Sei Π ein $\mathcal{NP}$-schweres Optimierungsproblem mit

- $\mathrm{OPT}(I) \in \mathbb{N}$ für alle $I \in D_\Pi$, und
- es existiert ein Polynom q mit $\mathrm{OPT}(I) < q(|I|)$ für alle $I \in D_\Pi$.

Falls $\mathcal{P} \neq \mathcal{NP}$, so gibt es kein FPTAS $\{\mathcal{A}_\varepsilon \mid \varepsilon > 0\}$ für Π.

- Für die Güte beobachte: $\mathrm{OPT}(I) \leq (1 + \varepsilon_0)\mathcal{A}_{\varepsilon_0}(I)$ und $\mathrm{OPT}(I) < q(|I|) = \frac{1}{\varepsilon_0}$.
- Also gilt $0 \leq \mathrm{OPT}(I) - \mathcal{A}_{\varepsilon_0}(I) \leq \varepsilon_0 \cdot \mathcal{A}_{\varepsilon_0}(I) \leq \varepsilon_0 \cdot \mathrm{OPT}(I) < 1$.
- Da $\mathrm{OPT}(I), \mathcal{A}_{\varepsilon_0}(I) \in \mathbb{N}$, ist $\mathrm{OPT}(I) = \mathcal{A}_{\varepsilon_0}(I)$.
- Demnach ist $\mathcal{A}(I) = \mathcal{A}_{\varepsilon_0}(I) = \mathrm{OPT}(I)$, also $\Pi \in \mathcal{P}$.
- Da Π $\mathcal{NP}$-schwer ist, folgt $\mathcal{P} = \mathcal{NP}$.

Übersicht


relative Approximation
$\mathcal{A}(I) \leq K \cdot \mathrm{OPT}(I)$ bzw. $\mathcal{R}^\infty_{\mathcal{A}} \leq K$
poly($|I|$)
min-Metric-TSP
PTAS
$\mathcal{A}_\varepsilon(I) \leq (1+\varepsilon) \cdot \mathrm{OPT}(I)$
poly($|I|$)
min-Vertex-Color
FPTAS
$\mathcal{A}_\varepsilon(I) \leq (1+\varepsilon) \cdot \mathrm{OPT}(I)$
poly($|I|, 1/\varepsilon$)
min-Edge-Color
absolute Approx.
$|\mathcal{A}(I) - \mathrm{OPT}(I)| \leq K$
poly($|I|$)
Klasse $\mathcal{P}$
$\mathcal{A}(I) = \mathrm{OPT}(I)$
poly($|I|$)
max-Knapsack

Approximierbarkeit von min-VERTEX-COLOR

Optimierungsproblem min-VERTEX-COLOR

Gegeben: Graph $G = (V, E)$

Aufgabe: Färbe die Knoten in V mit möglichst wenig Farben, so dass je zwei adjazente Knoten verschiedene Farben besitzen.

Satz.

Falls $\mathcal{P} \neq \mathcal{NP}$, dann existiert kein relativer Approximationsalgorithmus $\mathcal{A}$ für min-VERTEX-COLOR mit $\mathcal{R}_{\mathcal{A}}^{\infty} < \frac{4}{3}$.

Approximierbarkeit von min-VERTEX-COLOR

Optimierungsproblem min-VERTEX-COLOR

Gegeben: Graph $G = (V, E)$

Aufgabe: Färbe die Knoten in V mit möglichst wenig Farben, so dass je zwei adjazente Knoten verschiedene Farben besitzen.

Satz.

Falls $\mathcal{P} \neq \mathcal{NP}$, dann existiert kein relativer Approximationsalgorithmus $\mathcal{A}$ für min-VERTEX-COLOR mit $\mathcal{R}_{\mathcal{A}}^{\infty} < \frac{4}{3}$.

- Angenommen es gibt einen relativen Approximationsalgorithmus $\mathcal{A}$ für min-VERTEX-COLOR mit $\mathcal{R}_{\mathcal{A}}^{\infty} < \frac{4}{3}$.
- Wir benutzen $\mathcal{A}$ um Entscheidungsproblem 3COLOR zu lösen.
- Da 3COLOR $\mathcal{NP}$-schwer ist, folgt $\mathcal{P} = \mathcal{NP}$.

Approximierbarkeit von min-VERTEX-COLOR

Zu zwei Graphen $G_1 = (V_1, E_1)$ und $G_2 = (V_2, E_2)$ sei

$$G := (V, E) := G_1[G_2]$$

definiert durch

$$V := V_1 \times V_2$$

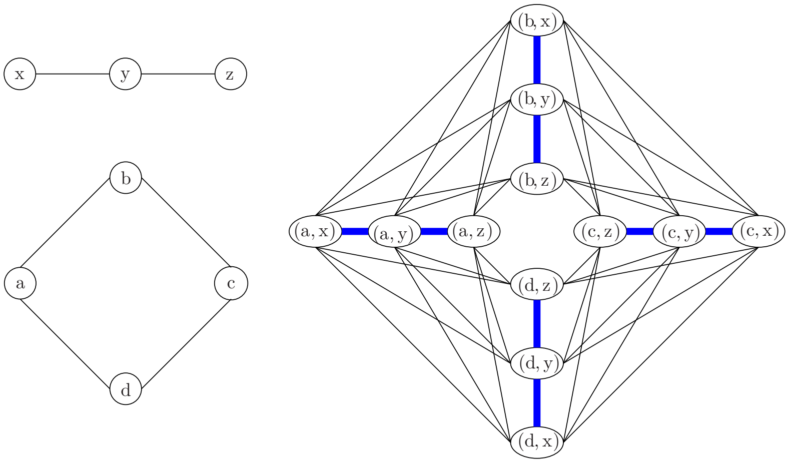
und

$$E := \left\{ \{(u_1, u_2), (v_1, v_2)\} \;\middle|\; \begin{array}{l} \text{entweder } \{u_1, v_1\} \in E_1, \text{ oder} \\ u_1 = v_1 \text{ und } \{u_2, v_2\} \in E_2 \end{array} \right\}$$

Anschaulich

- Jeder Knoten aus G_1 wird durch eine Kopie von G_2 ersetzt
- Jede Kante aus E_1 durch einen vollständig bipartiten Graphen zwischen den entsprechenden Kopien.

Approximierbarkeit von min-VERTEX-COLOR



Approximierbarkeit von min-VERTEX-COLOR

$$\mathcal{R}_{\mathcal{A}}^{\infty} := \inf \left\{ r \geq 1 \mid \begin{array}{l} \text{es gibt ein } N_0 > 0 \text{, so dass } \mathcal{R}_{\mathcal{A}}(I) \leq r \\ \text{für alle } I \text{ mit } \mathrm{OPT}(I) \geq N_0 \end{array} \right\}$$

- Angenommen es gibt einen relativen Approximationsalgorithmus $\mathcal{A}$ für min-VERTEX-COLOR mit $\mathcal{R}_{\mathcal{A}}^{\infty} < \frac{4}{3}$.
- Dann existiert ein $N_0 \in \mathbb{N}$ so, dass $\mathcal{A}(G) < \frac{4}{3}\,\mathrm{OPT}(G)$ für alle Graphen G mit $\mathrm{OPT}(G) \geq N_0$.

Approximierbarkeit von min-VERTEX-COLOR

- Dann existiert ein $N_0 \in \mathbb{N}$ so, dass $\mathcal{A}(G) < \frac{4}{3}\,\mathrm{OPT}(G)$ für alle Graphen G mit $\mathrm{OPT}(G) \geq N_0$.
- Sei also $G = (V, E)$ eine beliebige Instanz von 3COLOR.
- Dann definiere $G^* := K_{N_0}[G]$, wobei K_{N_0} der vollständige Graph mit N_0 Knoten ist.
- Dann gilt: $\mathrm{OPT}(G^*) = N_0 \cdot \mathrm{OPT}(G) \geq N_0$.

Fallunterscheidung:

- Falls G Ja-Instanz (also dreifärbbar) ist, gilt:
$$\mathcal{A}(G^*) < \frac{4}{3}\,\mathrm{OPT}(G^*) = \frac{4}{3} \cdot N_0 \cdot \mathrm{OPT}(G) \leq \frac{4}{3} \cdot N_0 \cdot 3 = 4N_0.$$
- Andererseits, falls G Nein-Instanz (also nicht dreifärbbar) ist, gilt
$$\mathcal{A}(G^*) \geq \mathrm{OPT}(G^*) = N_0 \cdot \mathrm{OPT}(G) \geq 4N_0.$$

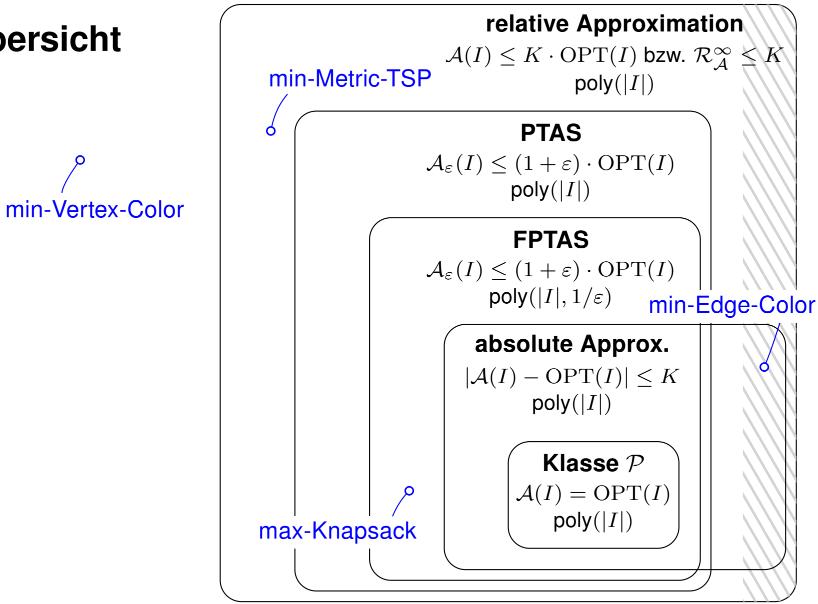
Fazit: G ist Ja-Instanz (dreifärbbar) genau dann, wenn $\mathcal{A}(G^*) < 4N_0$.

Approximierbarkeit von min-VERTEX-COLOR

Fazit: G ist Ja-Instanz (dreifärbbar) genau dann, wenn $\mathcal{A}(G^*) < 4N_0$.

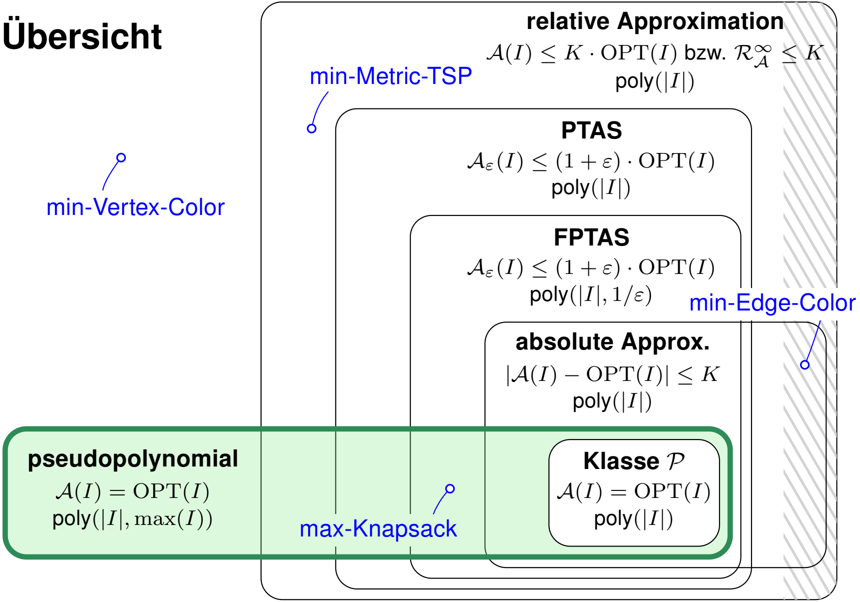
- Die Größe von G^* ist polynomial in der Größe von G.
- Also kann G^* in polynomialer Zeit konstruiert werden.
- Damit ist die Anwendung von $\mathcal{A}$ auf G^* polynomial in der Größe von G.
- Also haben wir einen polynomialen Algorithmus zur Lösung von 3COLOR konstruiert.
- Da 3COLOR $\mathcal{NP}$-schwer ist, folgt damit dass $\mathcal{P} = \mathcal{NP}$.

Übersicht



relative Approximation
$\mathcal{A}(I) \leq K \cdot \mathrm{OPT}(I)$ bzw. $\mathcal{R}_{\mathcal{A}}^{\infty} \leq K$
poly($|I|$)
min-Metric-TSP
PTAS
$\mathcal{A}_\varepsilon(I) \leq (1+\varepsilon) \cdot \mathrm{OPT}(I)$
poly($|I|$)
min-Vertex-Color
FPTAS
$\mathcal{A}_\varepsilon(I) \leq (1+\varepsilon) \cdot \mathrm{OPT}(I)$
poly($|I|, 1/\varepsilon$)
min-Edge-Color
absolute Approx.
$|\mathcal{A}(I) - \mathrm{OPT}(I)| \leq K$
poly($|I|$)
Klasse $\mathcal{P}$
$\mathcal{A}(I) = \mathrm{OPT}(I)$
poly($|I|$)
max-Knapsack

Übersicht



relative Approximation
$\mathcal{A}(I) \leq K \cdot \mathrm{OPT}(I)$ bzw. $\mathcal{R}_{\mathcal{A}}^{\infty} \leq K$
poly($|I|$)
min-Metric-TSP
PTAS
$\mathcal{A}_{\varepsilon}(I) \leq (1+\varepsilon) \cdot \mathrm{OPT}(I)$
poly($|I|$)
min-Vertex-Color
FPTAS
$\mathcal{A}_{\varepsilon}(I) \leq (1+\varepsilon) \cdot \mathrm{OPT}(I)$
poly($|I|, 1/\varepsilon$)
min-Edge-Color
absolute Approx.
$|\mathcal{A}(I) - \mathrm{OPT}(I)| \leq K$
poly($|I|$)
pseudopolynomial
$\mathcal{A}(I) = \mathrm{OPT}(I)$
poly($|I|, \max(I)$)
max-Knapsack
Klasse $\mathcal{P}$
$\mathcal{A}(I) = \mathrm{OPT}(I)$
poly($|I|$)

Ein allgemeines Resultat

Satz.

Sei Π ein Optimierungsproblem für das gilt:

- $\mathrm{OPT}(I) \in \mathbb{N}$ für alle $I \in D_\Pi$
- es existiert ein Polynom q mit $\mathrm{OPT}(I) \leq q(|I| + \max(I))$

Falls Π ein FPTAS hat, so hat es einen pseudopolynomialen optimalen Algorithmus.

Ende des Kapitels

- Wir haben heute das Kapitel **Komplexitätstheorie** abgeschlossen.
- Wir werden aber nochmal über Turing-Maschinen sprechen.

Testen Sie sich:

Können Sie mit folgenden Begriffen etwas anfangen?

CLIQUE polynomiale Transformation 3SAT

Zeitkomplexitätsfunktion Turing-Maschine Approximation

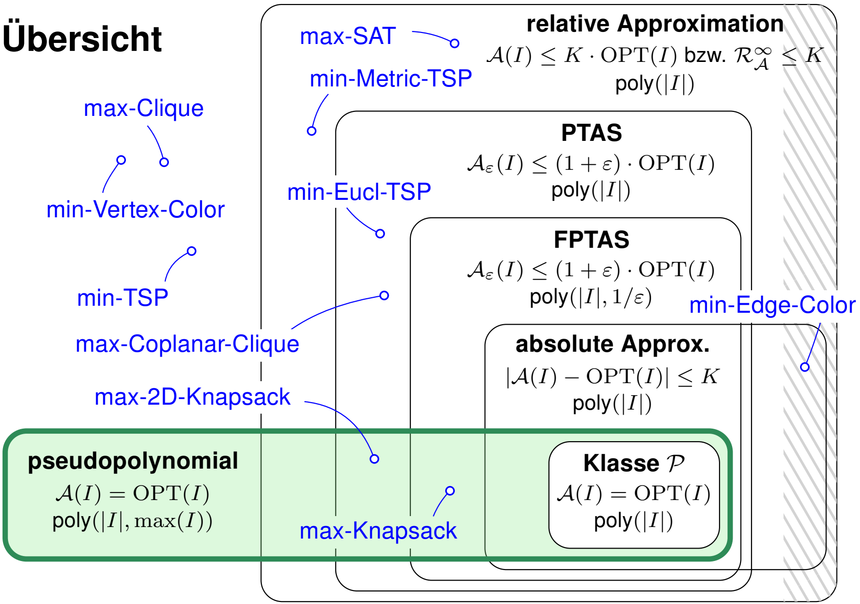
$\mathcal{NP}$ Orakel Eingabekodierung

Optimierungsproblem

Instanz pseudopolynomial $\mathcal{NP}$-vollständig (F)PTAS

Entscheidungsproblem $\mathcal{P}$ Nichtdeterminismus

Übersicht



relative Approximation
$\mathcal{A}(I) \leq K \cdot \mathrm{OPT}(I)$ bzw. $\mathcal{R}^\infty_\mathcal{A} \leq K$
poly$(|I|)$
PTAS
$\mathcal{A}_\varepsilon(I) \leq (1+\varepsilon) \cdot \mathrm{OPT}(I)$
poly$(|I|)$
FPTAS
$\mathcal{A}_\varepsilon(I) \leq (1+\varepsilon) \cdot \mathrm{OPT}(I)$
poly$(|I|, 1/\varepsilon)$
absolute Approx.
$|\mathcal{A}(I) - \mathrm{OPT}(I)| \leq K$
poly$(|I|)$
Klasse $\mathcal{P}$
$\mathcal{A}(I) = \mathrm{OPT}(I)$
poly$(|I|)$
pseudopolynomial
$\mathcal{A}(I) = \mathrm{OPT}(I)$
poly$(|I|, \max(I))$
max-SAT
min-Metric-TSP
max-Clique
min-Vertex-Color
min-Eucl-TSP
min-TSP
max-Coplanar-Clique
max-2D-Knapsack
min-Edge-Color
max-Knapsack